Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Environmental Sciences 11 (2011) 538 – 544

Procedia Environmental Sciences

Parameter Selection Algorithm for Support Vector Machine

Shuzhou Wang,Bo Meng

School of Electrical Engineering and Automation
Tianjin Polytechnic University ,Tianjin, China,,seek2000@163.com,bomeng@163.com

Abstract

Support Vector Machine (SVM) is a new modeling method. It has shown good performance in many field and mostly outperformed neural networks. The parameter selection should to be done before training SVM. Modified particle swarm optimization (POS) was adpoted to select parameters of SVM. It is shown by simulation that the modified POS algorithm can derive a set of optimal parameters of SVM. Compared with neural networks, SVM model possess some advantages such as simple structure, fast convergence speed with high generalization ability.

Keywords: support vector machine, parameter selection, chaotic particle swarm optimization

1.Introduction

Statistical Learning Theory focuses on the machine learning theory for small samples[1]. The main idea of the theory is to control the generalization ability of learning machine by controlling the complexity of its models. Support Vector Machine (SVM) is a new general machine learning algorithm developed from Statistical Learning Theory. It mostly outperformed traditional methods in theory and practice. SVM solved the problems such as small samples, high dimensions, nonlinear and local minimum problem. It has shown good performance in many fields such as pattern recognition and regression[2].

There are many kinds of function can be used for kernel of SVM, such as Gaussian and polynomial kernels. Wavelet function is a set of bases that can approximate arbitrary functions in arbitrary precision. So it is valuable to construct Wavelet SVM (WSVM) using a wavelet kernel[3]. First we construct a translation-invariant wavelet kernel by Marr wavelet.

Training SVM can be formulated as a quadratic programming problem. The parameter selection of SVM should to be done before resolving the QP problem. Particle swarm optimization (POS)[4]algorithm was adpoted to select parameters of SVM in this paper. To improve its global search ability, POS algorithm was modified by virtue of chaotic motion with sensitive dependence on initial conditions and ergodicity.

doi:10.1016/j.proenv.2011.12.085

2.Support vector machine

Support Vector Machine (SVM) was first used to pattern classification, and the basic idea is: mapping the data in input space with nonlinear transform $\phi(\cdot)$ to a high dimensional feature space, in which the problems become seeking for optimal linear classification hyper-plane. Similar to pattern classification, the basic idea of SVM for Regression (SVR) is: mapping the data in input space with nonlinear transform $\phi(\cdot)$ to a high dimensional feature space, and use linear function $f(x) = w^T\phi(x) + b^*$ to fit the sample data in the feature space, as well as ensure better generalization performance. Assume $x_i \in R^n, y_i \in R, \quad i = 1,\cdots,l$ as observation sample, R^n represents input space. SVR can be scribed as optimization problem of linearly constrained quadratic programming:

$$\min[\frac{1}{2}\| w \|^2 + \frac{C}{l}\sum_{i=1}^{l}(\zeta_i + \zeta_i^*)]$$
$$s.t.\begin{cases}((w\cdot x_i)+b) - y_i \le \varepsilon + \zeta_i \\ y_i - ((w\cdot x_i)+b) \le \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* \ge 0\end{cases} \tag{1}$$

Constant $C > 0$ is a tradeoff between function complexity and loss error. According KKT condition of optimization theory, derivative of Lagrange function of optimization problem (1) to variable w, b, ζ_i, ζ_i^* should be zero. Together with dual principle and kernel theory, the dual problem of optimization problem (1) can be obtained:

$$\min \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)$$
$$+\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \tag{2}$$
$$s.t. \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \quad 0 \le \alpha_i, \alpha_i^* \le C \quad i = 1,\cdots,l$$

$K(X, X')$ is kernel function, it takes the formation: $K(X, X') = \phi(X)\cdot\phi(X') = K(< X \cdot X' >)$ $X, X' \in R^n$. Mercer theorem[1] gives the conditions that a dot product kernel must satisfy. A translation invariant kernels, i.e. $K(X, X') = K(X - X')$, is an admissible SVM kernel if and only if the Fourier transform is non-negative[5]:

$$F[K](\omega) = (2\pi)^{-\frac{N}{2}}\int_{R^N}\exp(-j(\omega\cdot X))K(X)dX \ge 0 \tag{3}$$

This gives SVM kernel a necessary and sufficient condition for translation invariant kernels. It is useful for both checking if a function is an admissible kernel and actually constructing new kernels. The regression estimates for function with kernel are linear and takes the following form:

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_i, x) + b^* \tag{4}$$

Learning machine for regression function described above is support vector machine.

3.Wavelet function

A function $\psi(x)$ is called mother wavelet function if the function satisfies the admissible condition:

$$W_h = \int_0^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{5}$$

where $\Psi(\omega)$ is Fourier transform of $\psi(x)$. A function $f(x)$ can be approximated by a family of wavelet base functions $\psi_{a,b}(x)$, which is generated by dilations and translations of a mother wavelet function $\psi(x)$:

$$\psi_{a,b}(x) = |a|^{-\frac{1}{2}} \psi(\frac{x-b}{a}) \tag{6}$$

$x, a, b \in R$, and a, b is a factor of dilation and translation respectively. Therefore the wavelet transform and reconstruction of a function $f(x) \in L_2(R)$ can be written respectively as follow:

$$W_{a,b}(f) = < f(x), \psi_{a,b}(x) > \tag{7}$$

$$f(x) = \frac{1}{W_\psi} \int_{-\infty}^{\infty} \int_0^{\infty} W_{a,b}(f) \psi_{a,b}(x) \frac{1}{a^2} da db \doteq \sum_{i=1}^{l} W_i \psi_{a_i, b_i}(x) \tag{8}$$

In the expression (8), $\doteq$ means to approximate $f(x)$ in finite terms. One function that satisfies the admissible condition (5) is Marr wavelet function[6]:

$$\psi(t) = (1 - t^2) e^{-t^2/2} \tag{9}$$

First we construct a one-dimension translation- invariant wavelet kernel by Marr wavelet (9):

$$\begin{aligned} k(x, \bar{x}) &= k(x - \bar{x}) = \psi(\frac{x - \bar{x}}{a}) \\ &= (1 - (\frac{x - \bar{x}}{a})^2) \exp(-(\frac{x - \bar{x}}{a})^2 / 2) \end{aligned} \tag{10}$$

Where $x, \bar{x} \in R$. If tensor product of one-dimensional base function is used as a base of n-dimensional space, the kernel generating n-dimensional span space will be product of n one-dimension kernel[1]. Then we can directly obtain n-dimensional wavelet kernel according one-dimensional wavelet kernel (10):

$$\begin{aligned} K(X, \bar{X}) &= \prod_{j=1}^{N} k(x^j - \bar{x}^j) = \prod_{j=1}^{N} \psi(\frac{x^j - \bar{x}^j}{a}) \\ &= \prod_{j=1}^{N} (1 - \frac{(x^j - \bar{x}^j)^2}{a^2}) \exp(-\frac{(x^j - \bar{x}^j)^2}{2a^2}) \end{aligned} \tag{11}$$

4.Particle swarm optimization and chaotic particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation technique. It finds global optimum solution in search space through the interactions of individuals in a swarm of particles. Each particle represents a candidate solution to the problem and it has its own position and velocity. Suppose particle swarms are in D dimension search space, the particles can change velocities and positions using the following rules:

$$\begin{cases} v_{i,d}^{k+1} = \omega v_{i,d}^k + c_1 \cdot rand \cdot (p_{i,d}^k - x_{i,d}^k) + c_2 \cdot rand \cdot (p_{g,d}^k - x_{i,d}^k)) \\ x_{i,d}^{(k+1)} = x_{i,d}^{(k)} + v_{i,d}^{(k+1)} \end{cases} \tag{12}$$

where $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$, $v_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D})$, $p_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,D})$ denote the position, velocity, best personal positions of *i*-th particle respectively. $p_g = (p_{g,1}, p_{g,2}, \cdots, p_{g,D})$ denotes the best position

among all of the personal best positions achieved so far, $i = 1,2,\cdots,m$, i denote the index of particles, m denotes the number of particles in the swarm. i denotes the index of particles, $d = 1,2,\cdots,D$, denote the index of dimension of position and velocity. The superscripts k and $k+1$ denote the time index of the current and the next iterations respectively. Parameter ω is inertia weight factor that usually decreases linearly from 0.9 to 0.2 over the course of the run. The parameters $c1$ and $c2$ are called acceleration constant in interval of $(0,2)$ that adjust the maximum step of the particle flight toward p_i and p_g position. In order to reduce the likelihood of the particle leaving the search space, the value of each dimension of the velocity $v_{i,d}^k$ is clamped to the range $[-v_d^{\max}, v_d^{\max}]$. The value of $v_n^{\max}$ is usually chosen to be $v_d^{\max} = \overline{k} \cdot x_d^{\max}$, $0.1 \le \overline{k} \le 0.5$, where $x_d^{\max}$ is the upper bound of search region in the *n*-th dimension.

Maintain the position iterations expression in expression (12) and modify the velocity iterations expression, we can derive the compression factor model which can guarantee convergence easily[4]:

$$\begin{cases} v_{i,d}^{k+1} = \chi(v_{i,d}^k + \varphi_1 \cdot rand \cdot (p_{i,d}^k - x_{i,d}^k) \\ \qquad\qquad + \varphi_2 \cdot rand \cdot (p_{g,d}^k - x_{i,d}^k)) \\ x_{i,d}^{(k+1)} = x_{i,d}^{(k)} + v_{i,d}^{(k+1)} \end{cases} \tag{13}$$

where χ is convergence factor, whose value is subject to the function of $\varphi = \varphi_1 + \varphi_2$,and $\varphi_1 = \varphi_2 = 2.05$, so $\varphi = 4.1$, $\chi = 0.729$. Compared the two iterations expression, it can be seen that they are identical in case of $\omega = \chi = 0.729$, $c_1 = c_2 = \chi \cdot \varphi_1 = 1.49445$. However, the standard PSO algorithm has also some disadvantages like premature convergence phenomenon. In this paper POS algorithm was modified by virtue of chaotic motion.

Chaos is behaves of intrinsic stochastic process in nonlinear deterministic system. Chaos can track any state in a certain scope without repetition according to its regularity, and have the property of stochastic, ergodic and sensitivity to initial value. So it can be used to a method of reset position of initial particle and particle which arrive bound of search space. To generate chaotic variable, we select Logistic:

$$c_j^{r+1} = \mu c_j^r (1 - c_j^r) \quad r = 1,2,\cdots \tag{14}$$

where μ is control parameter, c_j is chaotic variable and $0 \le c_j \le 1$, $j = 1,2\cdots D$, D is dimension of particle position, r denote the index of variant particle needed to reset its initial position. when $\mu = 4$, the above expression is in entirely chaotic status , and variable c_j is ergodic in interval $(0,1)$. By virtue of character that chaotic motion is sensitive to initial values of, take D different initial values in interval $(0,1)$ except 0, 0.25, 0.50, 0.75, 1, put it to the above expression, and we derive D chaotic variable $c_1^1,\cdots,c_D^1$, corresponding to position of the first variant particle. The position of second variant particle is $c_1^2,\cdots,c_D^2$, and so on. In addition, it is need to map the chaotic variable c_j^r to variable x_j^r in optimization space:

$$x_j^r = x_j^{\min} + (x_j^{\max} - x_j^{\min}) c_j^r \tag{15}$$

Where $x_j^{\min}$, $x_j^{\max}$ is boundary value of x_j^r.

The main idea of modified PSO is: by virtue of chaotic motion with sensitive dependence on initial conditions and ergodicity, position of initial particle was initialized. Particle who reach

band of search space were put into the space again, whose position was initialized by the same chaotic optimization method.

We select parameter for WSVM with above modified PSO. Parameter ε controls the sparse of support vector of SVM and affects precision of the regression model. But it can be determined in practical application. So we only look kernel width parameter a and regularization coefficient C as optimizing target variable. Dimension of particle x_i is $D=2$. We define position vector x of PSO is $x_i=[a_i \ C_i,]$, and its initialization value is in the solution interval because of real-number code. $k-\text{fold}$ cross validation error was used as fitness function of PSO. Calculate j-th generalization error by the next expression:

$$e_j=\sqrt{\frac{1}{l_j}\sum_{i=1}^{l_j}(\hat{y}(x_i)-y(x_i))^2} \tag{16}$$

where $\hat{y}(x_i)$ is the forecast value for output of WSVM, and $y(x_i)$ is target output value, and l_j denote the number of *j*-th sample subset, $j=1,\cdots,k$. Repeat the process k times, and after k times iteration the average value of five generalization error is $k-\text{fold}$ cross validation error. Here let $k=5$. The best position among all of the personal best positions in the whole particle swarms represents a set of optimal parameters of WSVM. The pseudo-code for selecting parameters of WSVM with PSO can be described as follows:

Step1: Initialization: particle number of particle swarm m, banded value of position and velocity, $x_{\min},x_{\max}$, $v_{\min},v_{\max}$ iteration times M. And initialize initial position and velocity $x_i^0=\{a_i^0,C_i^0\}, i=1,\cdots,m$, v_i^0 by chaotic variables according expression (15), initial particle velocity v_i^0 randomly.

Step2: Separate the whole train samples even to 5 subset $S_1,\cdots S_5$ which contain no same samples each other, and let $k=0$.

Step3: Train SVM by optimal parameters to present $x_i^k=[a_i^k \ C_i^k]$, calculate $5-\text{fold}$ cross validation error $\overline{e}^k$:

Step3.1: initialize $j=1$.

Step3.2: subset S_j as test set, else as train set.

Step3.3: calculate generalization error e_j^k of S_j by expression (16), let $j=j+1$ repeat Step3.2 until $j=5$.

Step3.4: calculate average value of five generalization error, and derive $5-\text{fold}$ cross validation error $\overline{e}^k$.

Step4: Let $\overline{e}^k$ be value of fitness, denote personal and global best positions $\overline{e}^k$ as p_i, p_g respectively, let $k=k+1$, update position and velocity of particle according expression (13): $x_i^k=[a_i^k \ C_i^k]$, v_i^k.

Step5: If the position of particle arrive $x_{\min},x_{\max}$, initialize its position by chaotic variable according expression (15), and restrict the velocity of particle in interval $[v_{\min},v_{\max}]$.

Step6: repeat Step3, until maximum number of iteration $k=M$.

At last, form p_g can derive global optimal value:

$x=[a, \ C]$

5.Simulation experiment

Now, we validate the performance of wavelet kernel and PSO algorithm by simulation experiments. It is to approximate a two-variable function:

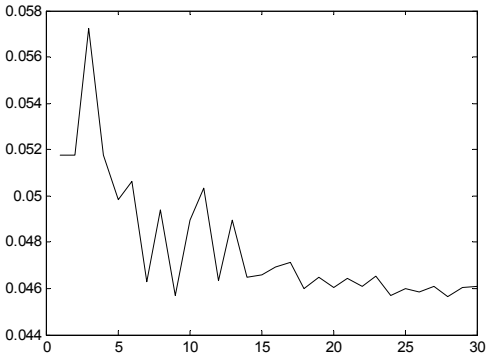
$$f(\boldsymbol{x}) = (x_1^2 - x_2^2)\sin(0.5x_1) \quad \boldsymbol{x} = \in [-10,10]\times[-10,10] \tag{17}$$

We take 400 points as he training examples, and 400 points as the testing examples. Pentium - IV 2.66G CPU, 256M memory and Matlab6.5 server as simulation platform. For comparison, we select another SVM with Gauss kernel, which marked as GSVM model. Two models are trained using the same training and testing samples. We resolve optimization problems of SVM with Sequential Minimal Optimization (SMO) algorithm [7,8].

Some empirical parameters need to be selected beforehand: $\varepsilon = 0.05$, target error $e = 0.001$, which marked as WSVM model. In the process of parameter selection for SVM, the search scope for width parameter kernel function and regularization coefficient were restrict in interval: $\sigma \in [0.001, 1]$, $C \in [1, 15]$, number of particle swarms $m = 20$. Let $v_d^{\max} = \overline{k} \cdot x_d^{\max}$, $\overline{k} = 0.2$, Maximum iteration times $M = 30$. All experiments consisted of 5 executions and the parameters with best fitness value were reserved. The result of optimization: $[\sigma \ C] = [2.50, 30]$ for Gauss kernel and $[a \ C] = [3.9, 30]$ for Marr kernel. Accuracy of train error and test error were measured by the following expression:

$$\delta = (\sum_{i=1}^{l}(y_i - f_i)^2)^{\frac{1}{2}} / (\sum_{i=1}^{l}(y_i - \overline{y})^2)^{\frac{1}{2}}, \quad \overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i \tag{18}$$

The relation of regularization coefficient and test error with number of iteration times were shown in Figure 1and Figure 2 respectively.



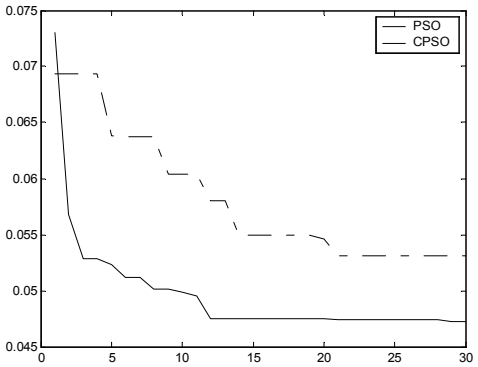


Fig.1 Convergence Process of a Single Particle for Optimization Fig.2 Comparison of Convergence of Two PSO algorithms

After training we can obtain WSVM and GSVM model. Then the test samples are input into the two trained model to validate their generalization performance. For comparison, we select BP neural network (BPNN) with 4 -10-1 structure, whose activation function of hidden and output layer are TANSIG, PURELIN respectively. It is trained with algorithm of gradient descent after proper parameter selection, which marked as BPNN model. Accuracy is measured by the Mean Square Error (MSE). Data of training time and accuracy for two models are shown in Table 1.

Table1. Performance Comparison of Three Model

	Training Time (Second)	Accuracy of Training	Accuracy of Test
WSVM	65.203	0.052	0.151
GSVM	76.439	0.061	0.192
NN	584	0.478	0.965

It can be seen from simulation results that wave SVM model has a good approximation to test samples.

6.Conclusion

Marr wavelet was used to construct wavelet kernel. Modified chaotic particle swarm optimization was adpoted to select parameters of SVM. It is shown by simulation that the CPOS algorithm can derive a set of optimal parameters of WSVM, and WSVM possess some advantages such as fast convergence speed and high generalization ability compared with Gauss kernel SVM. Compared with neural networks, SVM model possess some advantages such as simple structure, fast convergence speed with high generalization ability.

References

[1] Vapnik V., The Nature of Statistical Learning Theory. Springer Verlag, New York. 2000.

[2] M. Doumpos, C. Zopounidis. Additive Support Vector Machines for Pattern Classification. IEEE Trans. on Systems, Man, and Cybernetics, Part B, 2007,37(3): 540-550.

[3] Li Zhang, Weida Zhou, Licheng Jiao, Wavelet Support Vector Machine. IEEE transactions on Systems, man, and cybernetics, 2004,34(1): 34-39.

[4] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings of the Congress on Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1951-1957

[5] A. Smola, B. Schölkopf, and K.-R. Müller, The connection between regularization operators and support vector kernels. Neural Network, 1998,11(4): 637–649.

[6] S. Mallat. A Wavelet Tour of Signal Processing. China Machine Press, 2003

[7] Flake GW, Lawrence S, Efficient SVM regression training with SMO. Machine Learning Special Issue on SVMs, 2002,46(1-3): 271-290.

[8] Pai-Hsuen Chen, Rong-En Fan, Chih-Jen Lin, Study on SMO-type Decomposition Methods for SVM. IEEE transactions on neural networks 2006,17(4): 893-908